BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Penelitian

Pada penelitian sebelumnya yang berjudul "***Penerapan Analisis Sentimen pada Twitter Berbahasa Indonesia sebagai Pemberi Rating",*** penelitian ini berkaitan dengan data komentar pada *twitter* yang mengelompokkannya kedalam polaritas tertentu yaitu positif, negatif, dan netral akan tetapi keluaran tersebut kurang bermakna sehingga perlu diterjemahkan kedalam bentuk *rating* dengan rentang nilai 1 sampai 5 , karena *rating* merupakan indikator kesuksesan untuk menunjang promosi misalnya pada produk suatu perusahaan (Monarizqa, et al., 2014).

Penelitian lainnya dengan judul "***Predicting Star Rating of Movie Raview Comments",*** penelitian bertujuan memprediksi *rating* pada data komentar film dengan skala nilai *rating* 1 sampai 10 serta bertujuan untuk mengetahui pengaruh metadata seperti *director, actors, budget, production house* , dan masih banyak lagi yang ada pada komentar terkait *film* tersebut sehingga mempengaruhi *rating,* dengan penggabungan metode *naïve bayes* dan metadata hasil akurasi yang diperoleh 49.78% lebih rendah dibandingkan hanya menggunakan metode *naïve bayes* saja sebesar 51.23%, dikarenakan kurangnya wawasan penonton sendiri terhadap metadata dan penilaian mereka juga bervariasi tidak tergantung dengan metadata (Scaria, et al., 2011).

Pada penelitian ahmad mengenai ***"Penerapan Text Mining dalam Klasifikasi Judul Skripsi"*** bertujuan membuat model data judul skripsi di bidang teknik informatika sehingga dapat dikategorikan secara otomatis. Pengklasifikasian judul tersebut menggunakan metode SVM dan NBC dengan fitur *n-gram* (*unigram, bigram* dan *trigram*). Hasil dari penerapan metode klasifikasi menunjukkan metode NBC memperoleh akurasi tinggi dengan akurasi *unigram* 97%, *bigram* 97% dan *trigram* 98% dan SVM memperoleh akurasi untuk *unigram* 73%, *bigram* 59% dan *trigram* 60%. Metode SVM mendapatkan hasil rendah dibandingkan dengan NBC disebabkan pada penelitian tersebut menggunakan *linear kernel*. Penggunaan *linear kernel* sangat efektif sekali pada *binary classification* berbeda dengan penelitian ini yang menggunakan *multiclass* untuk klasifikasi judul skripsi dengan kelas Rekayasa Perangkat Lunak (SIRPL), Komputasi dan Sistem Cerdas (KSC), Grafika dan Multimedia (GFMD), serta Sistem dan Jaringan Komputer (SJK) (Hidayatullah, 2016).

Pada penelitian lain dengan judul ***"Penggunaan N-gram pada Analisa Sentimen Pemilihan Kepala Daerah Jakarta menggunakan Algoritma Naïve Bayes***" bertujuan untuk melihat hasil peningkatan akurasi dengan penggunaan *n-gram* meliputi *unigram, bigram* dan *trigram*. Penerapan *n-gram* disebabkan dalam bahasa Indonesia banyak *frase* yang tidak hanya berdiri sendiri dan diharapkan dengan *n-gram* dapat menangkap penggabungan kata sifat yang sering muncul untuk menunjukkan kelas *sentiment*. Hasil pengujian pada penelitian ini akurasi *n-gram* dengan nilai tinggi adalah *bigram*. Adapun *unigram* dan *bigram* performanya lebih baik dibandingkan dengan *trigram.* Hasil pengujian dari *unigram* untuk

precission 74.3%, *recall* 77.3% dan *accuracy* 78.5%. Hasil akurasi dari *bigram* untuk *precission* 76%, *recall* 88.9% dan *accuracy* 82.3%. Hasil akurasi dari *trigram* untuk *precission* 12.3%, *recall* 89.8% dan *accuracy* 54.7%. *Precission* dan *accuracy* dari *trigram* lebih rendah dibanding *unigram* dan *bigram*. Hal tersebut membuktikan *trigram* tidak begitu efektif disebabkan pengambilan tiga suku kata lebih cenderung pada dataset negatif (Indhiarta, 2017).

Pada penelitian lainnya yang berjudul "***Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming menggunakan Metode Naïve Bayes dan SVM"*** juga menggunakan fitur *n-gram* meliputi *unigram, bigram* dan *trigram.* Hasil akurasi tertinggi diperoleh dengan *pre-processing* normalisasi dan *stemming* untuk kedua metode. Hasil akurasi dari *unigram* dan gabungan *n-gram* lebih baik mendekati 90% dibandingkan dengan *bigram* dengan akurasi mendekati 70% dan *trigram* dengan akurasi mendekati 50% untuk kedua metode. Hasil menurun seiring dengan nilai *n* semakin naik sehingga frekuensi kata menurun. Hal tersebut berbeda dengan menggabungan *n-gram* dan *unigram* kemungkinan mendapatkan kata yang sama lebih besar (Saputra, et al., 2015)

2.2 *Text Mining*

Text mining merupakan proses untuk analisis teks untuk mengekstrak informasi serta bertugas untuk menemukan pola pada suatu teks untuk menghasilkan informasi baru pada tumpukan teks dalam jumlah besar, tentunya informasi baru tersebut berguna untuk bidang tertentu hampir mirip dengan data mining akan tetapi ada perbedaan diantara keduannya. Data mining yang lebih populer terlebih dahulu dibandingkan dengan *text mining*. Pada data mining ini mampu mengenali pola terhadap suatu data secara implisit dibandingkan dengan *text mining* informasi yang diekstrak haruslah jelas dan informasi tersebut secara eksplisit ada pada teks (Witten, 2004). Banyak sekali permasalahan yang berkaitan dengan *text mining* diantaranya adalah *natural language processing, information extraction, information retrieval, classification* dan *clustering* (Berry & Kogan, 2010).

2.3 *Pre-Processing*

Proses ini merupakan tahap awal dari sebuah pemrosesan teks. Hasil *pre-processing* akan digunakan untuk proses selanjutnya dalam penelitian ini akan digunakan dalam proses klasifikasi *rating*. Menurut Aris menjelaskan bahwa *pre-processing* adalah sebuah proses pengurangan kata-kata tidak penting, tidak mempunyai arti dari database teks atau dokumen, sehingga membuat data lebih terstruktur dan siap untuk diolah (Harjanta, 2015). *Pre-processing* pada penelitian ini untuk mendapatkan teks terstruktur meliputi proses *case folding, tokenizing, filtering, stemming dan n-gram.*

2.3.1 *Case Folding*

Proses *case folding* mengubah semua huruf kapital menjadi huruf kecil pada teks atau sebaliknya. Pada penelitian ini semua huruf diubah menjadi huruf kecil, lebih detailnya pada Tabel 2.1

Tabel 2.1 Ilustrasi *Case folding*

Masukan	Keluaran
Wanginya super enak, feminin banget gitu wanginya. BAGUS pokoknya ^_^	wanginya yang super enak, feminin banget gitu wanginya. bagus pokoknya ^_^

2.3.2 *Tokenizing*

Merupakan proses menghilangkan tanda baca, angka, dan karakter selain *alphabet*. Pada tahap ini juga dilakukan pemisahan kata sesuai dengan kata penyusunnya biasanya batas akhir dari kata tersebut ditandai dengan karakter spasi (Hartanti & Hartanti, 2016), lebih detailnya pada Tabel 2.2

Tabel 2.2 Ilustrasi *Tokenizing*

Masukan	Keluaran
wanginya yang super enak, feminin banget gitu wanginya. bagus pokoknya ^_^	wanginya super yang enak feminin banget gitu wanginya bagus pokoknya

2.3.3 *Filtering*

Proses *filtering* menggunakan hasil *tokenizing* untuk mengambil kata-kata penting. *Filtering* mempunyai dua *algoritme stoplist* menghilangkan kata-kata yang tidak penting dan *wordlist* mempertahankan kata-kata penting. Pada penelitian ini menggunakan *stoplist* atau *stopword removal* kata-kata yang dihilangkan merupakan kata umum yang sering muncul pada dokumen. Pada kasus IR atau *Information Retrieval* bahwa *stopword* ini berpengaruh pada kecocokan dokumen yang dibutuhkan oleh pengguna dikarenakan kata tersebut sering muncul dibanyak dokumen, misalnya kata hubung ‘dan’, ‘serta’, ‘demikian’, menghilangkan kata depan ‘di’, ‘ke’ (Manning, et al., 2009),dan masih banyak lagi kata-kata yang tidak berkaitan dengan topik pada dokumen yang sering muncul. Oleh karena itu, diperlukannya proses *filtering*. Kata adverbia dan kata negasi dianggap tidak penting termasuk pada kamus *stopword*. Pada penelitian ini kata tersebut dianggap penting sehingga keberadaannya pada dokumen tetap dipertahankan (Destuardi & Sumpeno, 2009). Kata adverbia merupakan kata-kata yang menjelaskan *verba, adjektiva,* atau adverbia lainnya contoh kata ‘amat’, ‘sangat’, ‘sekali’,dll, (Alwi, et al., 2010). Kata negasi menyebabkan perubahan polaritas dari

suatu pernyataan. Contoh negasi adalah kata ‘tidak’, ‘bukan’, ‘tanpa’ (Destuardi & Sumpeno, 2009). Contoh dari *stopword removal* pada Tabel 2.3

Tabel 2.3 Ilustrasi *Filtering*

Masukan	Keluaran
Wanginya super yang enak feminin banget gitu wanginya bagus pokoknya	wanginya super enak feminin banget gitu wanginya bagus pokoknya

2.3.4 *Stemming*

Stemming merupakan proses untuk mendapatkan kata dasar dengan cara menghilangkan imbuhan yang melekat pada kata tersebut. Pencarian kata dasar setiap bahasa berbeda-beda untuk bahasa indonesia proses *stemming* menghilangkan imbuhan berada di awal *preffix*, berada ditengah *inffix,* berada di akhir *suffix*, gabungan diawal dan diakhir *confix*, berbeda dengan bahasa lain seperti bahasa inggris pencarian kata dasar hanya dengan menghilangkan *suffix*. Ilustrasi pada Tabel 2.4 merupakan contoh proses dari *stemming*, penelitian ini menggunakan *stemming sastrawi* dan menggunakan *library sastrawi library* ini didesain dengan kualitas yang baik begitu pula dengan dokumentasinya (Librian, 2017). *Algoritme* pada *stemming sastrawi* meliputi Nazief andriani*, Confix Striping, Enhance Confix Striping, Modified Enhance Confix Striping* sehingga *stemming sastrawi* bekerja lebih baik. *Library* yang dipakai pada penelitian ini adalah *pysastrawi* merupakan *library sastrawi* untuk bahasa pemrograman *python*.

Tabel 2.4 Ilustrasi *Stemming*

Masukan	Keluaran
wanginya super enak feminin banget gitu wanginya bagus pokoknya	wangi super enak feminin banget gitu wangin bagus pokok

2.3.5 *N-gram*

Bahasa tidak terbentuk dari kata-kata individu, tetapi terdiri dari urutan kata individu dan frase 2, 3 atau lebih kata yang lebih dikenal *n*-gram dengan masing - masing kata tersebut mengandung informasi tersendiri, contoh penerapan *n-gram* khususnya *trigram* pada kalimat "*The cat sat on the mat*" menjadi "the cat sat", "*cat sat on*", "*sat on the*" dan "*on the mat*" apabila terdapat sebuah *punctuation* seperti koma, semi kolon, dan sebagainya maka pada proses *n-gram* tidak melewatinya melainkan membuat *n-gram* baru dengan kata setelah karakter tersebut misalnya pada kalimat *"Three blind mice, see how they run"* untuk *trigram* menjadi *"three blind mice", "see how they"* dan *"how they run"* (Ha, et al., 2003). Pada penelitian ini menerapkan *n-gram* dengan pemecahan kata pada kalimat ulasan meliputi *unigram* adalah pemecahan kata pada kalimat ulasan dengan *n*=1 atau *term* tunggal, *bigram* adalah pemecahan *n*-kata pada kalimat ulasan dengan *n*=2, dan kombinasi merupakan gabungan dari *unigram* dan *bigram*. Ilustrasi *n-gram* pada Tabel 2.5

2.4 Pembobotan *TF*

Term atau kata yang banyak muncul pada sebuah dokumen mempunyai nilai bobot yang tinggi merupakan jenis pembobotan kata *tf* atau *term frequency*. Pembobotan ini dilambangkan dengan $tf_{t,d}$ dimana t adalah *term* atau kata dan d adalah dokumen dengan menghitung jumlah kemunculan pada masing-masing dokumen dengan pendekatan *bag of words* yaitu mengabaikan urutan kata secara sintaksis hanya menekan pada menghitung kemunculan kata pada dokumen (Manning, et al., 2009).

2.5 *Naïve Bayes Classifier* (NBC)

Bayesian classifier lebih dikenal dengan *algoritme naïve bayes classifier* merupakan teknik klasifikasi dapat bekerja lebih cepat dengan akurasi yang tinggi pada jumlah data yang besar, NBC ini mengasumsikan bahwa efek dari nilai *attribut* pada kelas tertentu tidak tergantung pada nilai *attribut* lainnya. Asumsi tersebut disebut *class conditional independence*, untuk memudahkan perhitungan maka pengertian ini dianggap "*naïve*" (Han, et al., 2011), sederhananya yang dimaksud dengan *naïve* mengasumsikan bahwa kemunculan kata dalam suatu kalimat ulasan tidak dipengaruhi kata-kata lain pada kenyataannya kemungkinan kata dalam kalimat sangat dipengaruhi kemungkinan keberadaan kata-kata lain dalam kalimat tersebut (Destuardi & Sumpeno, 2009). Persamaan umum metode *naïve bayes* (Han, et al., 2011)

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{2.1}$$

Keterangan :

$P(C_i|X)$ = *posterior* merupakan peluang kelas C_i dimana *i* = 1,2,3…*m* dengan syarat X adalah *tuple* yang berisi sekumpulan atribut

$P(X|C_i)$ = *conditional probability* menghitung peluang *tuple* dengan syarat kelas tertentu

$P(C_i)$ = *prior* merupakan peluang kemuculan dokumen kelas ke *i* pada data latih yang terbebas dari atribut X

$P(X)$ = *prior probability* dari *X*

conditional probability mengasumsikan *attribut – attribut* pada *tuple* saling bebas tidak bergantung satu dengan lainnya, lebih detailnya untuk dijabarkan pada Persamaan 2.2 (Han, et al., 2011)

$$P(X|Ci) = \prod_{k=1}^{n} P(x_k|C_i)$$
$$= P(x_1|C_i) \text{ x } P(x_2|C_i) \text{ x } \ldots P(x_n|C_i) \quad (2.2)$$

Pada *prior probability* dari X bersifat konstan untuk semua kelas maka dihilangkan, sehingga menghasilkan persamaan baru (Han, et al., 2011)

$$P(C_i|X) = P(X|C_i)P(C_i) \quad (2.3)$$

Mengacu pada persamaan di atas maka untuk penelitian ini dengan objek ulasan produk kecantikan untuk X adalah *term* kata pada ulasan produk kecantikan sehingga persamaan menjadi (Manning, et al., 2009)

$$P(C_j|W_i) = P(C_j) \prod_{1 \le i \le nd} P(W_i|C_j) \quad (2.4)$$

Keterangan :

$P(C_j|W_i)$ = *posterior* merupakan menghitung peluang kemunculan kelas C_j dimana *j* = 1,2,3 . . . *m* dengan syarat W_i adalah kumpulan dari *term* kata dengan *i* = 1,2,3 . . .*n*

$P(W_i|C_j)$ = *conditional probability* adalah menghitung peluang kemunculan *term* kata pada kelas ke *j*. Perhitungan ini untuk semua *term* kata hasil *pre-processing* dari *token* kata ke *i* sampai *token* kata terakhir *nd*.

$P(C_j)$ = *prior* adalah menghitung peluang kemunculan dokumen kelas *j* pada data latih

Perhitungan peluang kemunculan dokumen kelas *j* pada data latih yang lebih dikenal dengan *prior* lebih detailnya pada Persamaan 2.5 (Manning, et al., 2009)

$$P(C_j) = \frac{N_{C_j}}{N} \tag{2.5}$$

Keterangan :

N_{C_j} = jumlah dokumen latih pada kelas *j*

N = jumlah seluruh dokumen latih

2.5.1 *Gaussian Naïve Bayes*

Tipe *gaussian naïve bayes* digunakan ketika nilai atribut pada X_i adalah *continue,* sehingga dalam perhitungan *conditional probability* melibatkan μc_i merupakan *mean* pada kelas ke *i* dan σc_i *standart deviation* pada kelas ke *i* (Han, et al., 2011).

2.5.2 *Bernoulli Naïve Bayes*

Tipe klasifikasi *bernoulli naïve bayes* menekankan pada ada atau tidak adanya *term* kata pada kelas dengan menggunakan *binary* yaitu 0 dan 1, untuk 0 apabila *term* kata tidak ada pada dokumen kelas tertentu dan 1 mengidikasikan bahwa *term* kata terdapat pada dokumen kelas tertentu (Manning, et al., 2009).

2.5.3 *Multinomial Naïve Bayes*

Tipe *multinomial naïve bayes* merupakan perhitungan bersifat *positional independence* tidak tergantung pada posisi atupun urutan kata. Pada prosesnya menghitung jumlah kata pada seluruh posisi kata pada dokumen, untuk persamaan *conditional probability* pada Persamaan 2.6 (Manning, et al., 2009)

$$P(W_i|C_j) = \frac{Count(W_i, C_j)}{((\sum_{w \epsilon V} Count(w, C_j) + |V|)} \tag{2.6}$$

Keterangan :

$P(W_i|C_j)$ = *conditional probability*

$Count(W_i, C_j)$ = menghitung jumlah kata W_i dengan *i* = 1,2,3.. *m* pada kelas *j* dengan *j* = 1,2,3,…*n*

$Count(w, C_j)$ = menghitung jumlah seluruh kata kelas *j*

$|V|$ = menghitung kata unik pada seluruh dokumen

untuk kondisi dimana suatu *term* kata tidak ditemukan pada data latih maka akan menghasilkan nilai *conditional probability* sama dengan nol sehingga perhitungan *posterior* juga menghasilkan nilai nol dengan hasil *posterior* nol dokumen gagal untuk diklasifikasikan sehingga untuk menghindari hal tersebut maka menambahkan satu biasanya dikenal dengan *Laplace smoothing*, dengan menambahkan satu mengasumsikan bahwa *term* kata paling tidak muncul satu kali

pada kelas tertentu pada data latih, sehingga persamaan yang didapat setelah menambahkan *Laplace smoothing* (Manning, et al., 2009):

$$P(W_i|C_j) = \frac{Count(W_i, C_j) + 1}{((\sum_{w \epsilon V} Count(w, C_j) + |V|)} \quad (2.7)$$

Tabel 2.5 Ilustrasi *N-gram*

Ulasan	*Unigram*	*Bigram*	Kombinasi
pelembab yang cocok buat kulit berminyak karena langsung menyerap dan tidak lengket.	pelembab yang cocok buat kulit berminyak karena langsung menyerap dan tidak lengket	pelembab yang yang cocok cocok buat buat kulit kulit berminyak berminyak karena kerena langsung langsung menyerap menyerap dan dan tidak tidak lengket	pelembab yang cocok buat kulit berminyak karena langsung menyerap dan tidak lengket pelembab yang yang cocok cocok buat buat kulit kulit berminyak berminyak karena kerena langsung langsung menyerap menyerap dan dan tidak tidak lengket

2.6 Pengujian

Proses pengujian merupakan kegiatan yang membandingkan antara hasil implementasi dengan kriteria standar yang telah ditetapkan untuk melihat keberhasilannya. Dari hasil pengujian akan dievaluasi sehingga tersedia informasi mengenai sejauh mana suatu kegiatan tertentu telah dicapai dan bisa diketahui

bila terdapat selisih antara standar yang telah ditetapkan dengan hasil yang bisa dicapai. Pengujian dilakukan pada penelitian ini membandingkan perhitung nilai dari hasil prediksi sistem dengan data uji kemudian menghitung akurasinya. Untuk menghitung nilai akurasi dapat menggunakan rumus Persamaan (Mahmudy & Widodo, 2014)

$$Akurasi = \frac{Jumlah\ data\ uji\ benar}{Jumlah\ seluruh\ data\ uji} * 100\% \quad (2.8)$$